Koble seg på fysisk på Tiago

Først kan du lage en ssh config-fil. OBS, funker ikke med eduroam, så først må du ssh deg inn på din bruker (kan også lage en config-fil dette):

```
$ ssh -L 8080:localhost:8080 <uiobrukernavn>@login.ifi.uio.no
```

Deretter kan du opprette en config-fil

```
$ cd
$ cd .ssh
$ touch config
$ vim config
```

Kopier inn dette:

```
Host tiago
  Hostname 193.157.209.134
  User pal
  IdentitiesOnly=yes
  IdentityFile ~/.ssh/tiago
```

Dette er en nokså ustabil kobling, for å gjøre den sikrere så kan du koble den opp med *'mosh'* i stedet for ssh.

Når du har satt opp tiago-config-filen kan du skrive:

```
$ ssh -L 8080:localhost:8080 tiago
```

Du er nå koblet på tiago. Om ikke en nettside dukker opp med kommandoen over kan du skrive inn dette i url-en:

```
http://localhost:8080/
```

Hvis du er koblet direkte opp med ethernet-kabel kan du skrive:

```
$ ssh -L 8080:localhost:8080 pal@10.68.0.1
```

Tips
For å få flere terminal-screens når du er ssh-et inn på tiago kan du bruke noe som heter *'screen'* (må lastes ned først). Dette er nyttig hvis du må kjøre roscore, roslaunch, osv.

Sende filer/folder over ssh

```
$ scp -r <source folderpath>
<brukernavn>@login.ifi.uio.no:~/<destination folderpath>
```

Videre så må du gjøre det samme i din eduroam videre til pal sitt nettverk

```
$ scp -r <source folderpath> pal@tiago:~/<destination folderpath>
```

Kjøre selvlagde programmer på Tiago
Se kap. 10.4 i Tiago-håndboken

Control

Joint GUI:

- rosrun rqt_joint_trajectory_controller rqt_joint_trajectory_controller

- Eller du kan kjøre tiago manuelt med kommandolinjen
  - `rosrun play_motion move_joint head_2_joint -0.6 2.0`
    - Her kan man bytte ut hvilken del/joint man vil flytte på
    - Koden over gir outputen:
      - `Moving joint head_2_joint to position -0.6 in 2.0s`

Look what Tiago is seeing:

- `roslaunch tiago_gazebo tiago_gazebo.launch robot:=steel public_sim:=true world:=look_to_point`
- `rosrun look_to_point look_to_point`
  - By clicking on any pixel of the image the robot will move its head in order to look at that point. With this we can easily lower or raise the head, look right or leftwards, etc.

Names of motions

- `rosparam list | grep  "play_motion/motions" | grep "meta/name" | cut -d '/' -f 4`
  - `close`
  - `close_half`
  - `do_weights`
  - `gun_hand`
  - `head_tour`
  - `home`
  - `inspect_surroundings`
  - `offer`
  - `open`
  - `pick_from_floor`
  - `pinch_hand`
  - `point`
  - `pregrasp_weight`
  - `prepare_grasp`
  - `reach_floor`
  - `reach_max`
  - `shake_hands`
  - `thumb_up_hand`
  - `unfold_arm`
  - `wave`

  Motions GUI

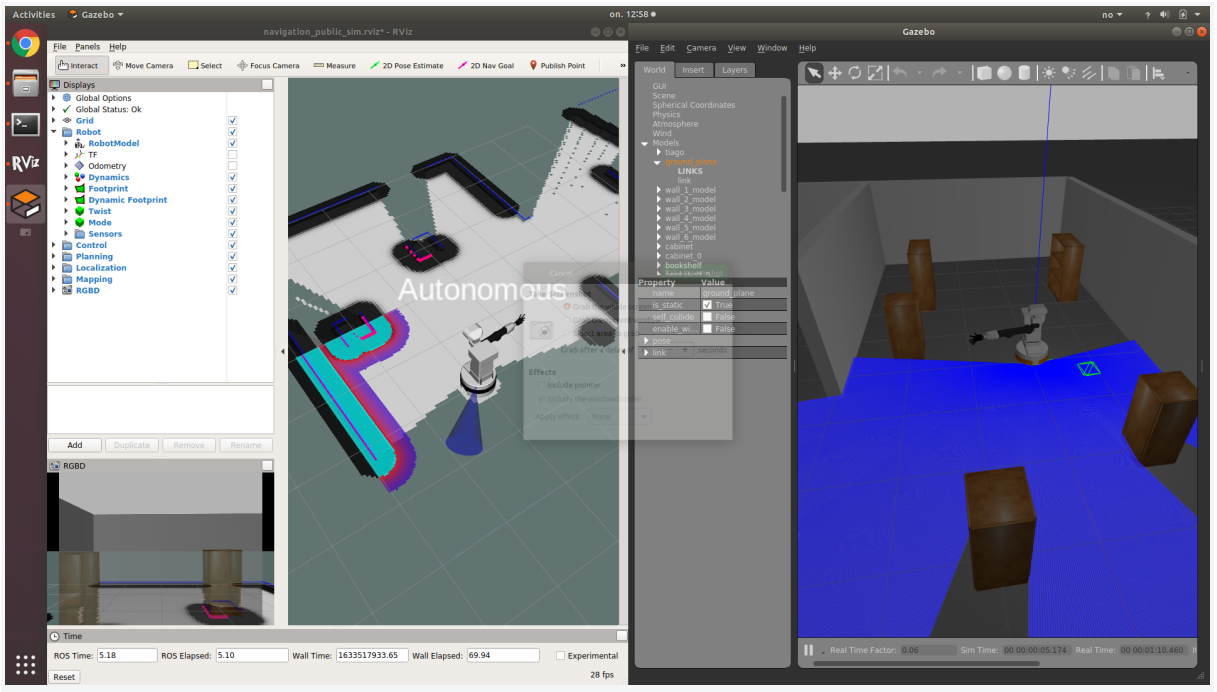
  - `rosrun actionlib axclient.py /play_motion`

- To run the different motion above, put motion in motion_name: '[motion name here]'
- Video of the different motions: https://www.youtube.com/watch?v=alV58zbfZOo

Navigation

Creating a map

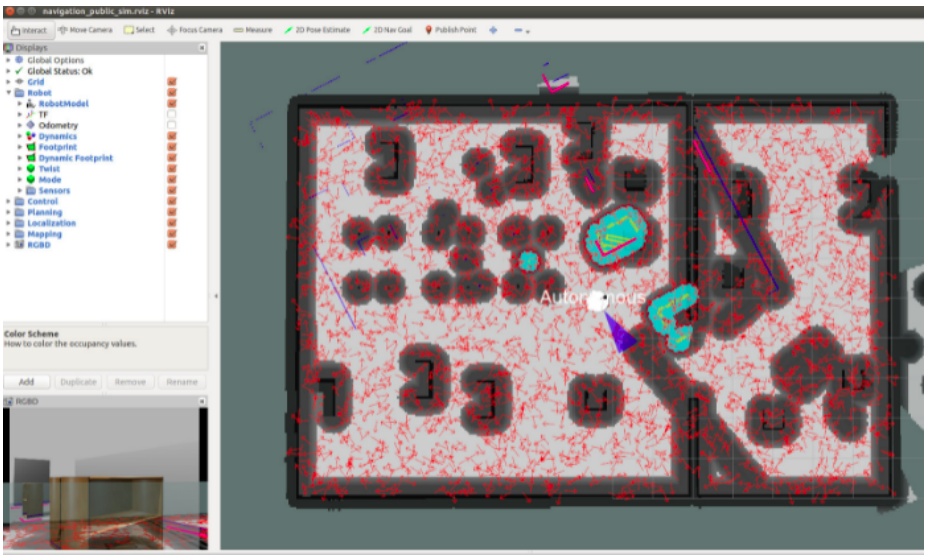
- `roslaunch tiago_2dnav_gazebo tiago_mapping.launch public_sim:=true`



- Dette vil launche gazebo og rviz (som gir mer info om navigeringen)
- Deretter kan du i et annet kommando-vindu kjøre:
  - `rosrun key_teleop key_teleop.py`
  - Som gjør at du kan navigere Tiago med piltaster
    - By pressing the arrow keys on this console drive TIAGo around the world. The map being created will be shown.
  - Press 'q' in the key_teleop console and save the map as follows
  - `rosservice call /pal_map_manager/save_map "directory: ''"`
  - The service call will save the map in the following folder
  - `cd ~/.pal/tiago_maps/config`

Navigation within a map

- `roslaunch tiago_2dnav_gazebo tiago_navigation.launch public_sim:=true lost:=true`
- In order to localize the robot run the following instruction in the second console:
  - `rosservice call /global_localization "{}"`

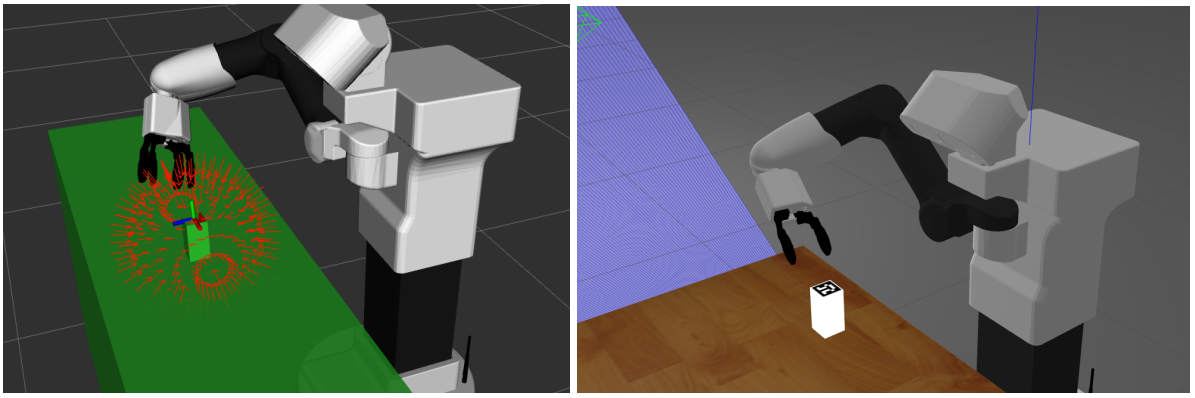


- Now a good way to help the particle filter to converge to the right pose estimate is to move the robot. A safe way to do so is to make the robot rotate about itself. You may use the left and right arrows of the key_teleop in order to do so.
    - `rosrun key_teleop key_teleop.py`
- Now, it is preferable to clear the costmaps as it contains erroneous data due to the bad localization of the robot:
    - `rosservice call /move_base/clear_costmaps "{}"`
- In order to send the robot to another location of the map the button 2D Nav Goal can be pressed in rviz.
    - Then, by clicking and holding the left button of the mouse over the point of the map at which we want to send the robot a green arrow will appear. By dragging the mouse around the point the orientation will change.

Octomap

- http://wiki.ros.org/Robots/TIAGo/Tutorials/MoveIt/Planning_Octomap

Pick & Place demo



- `roslaunch tiago_pick_demo pick_simulation.launch`
    - Will launch the a world with a table and an object
- `roslaunch tiago_pick_demo pick_demo.launch`

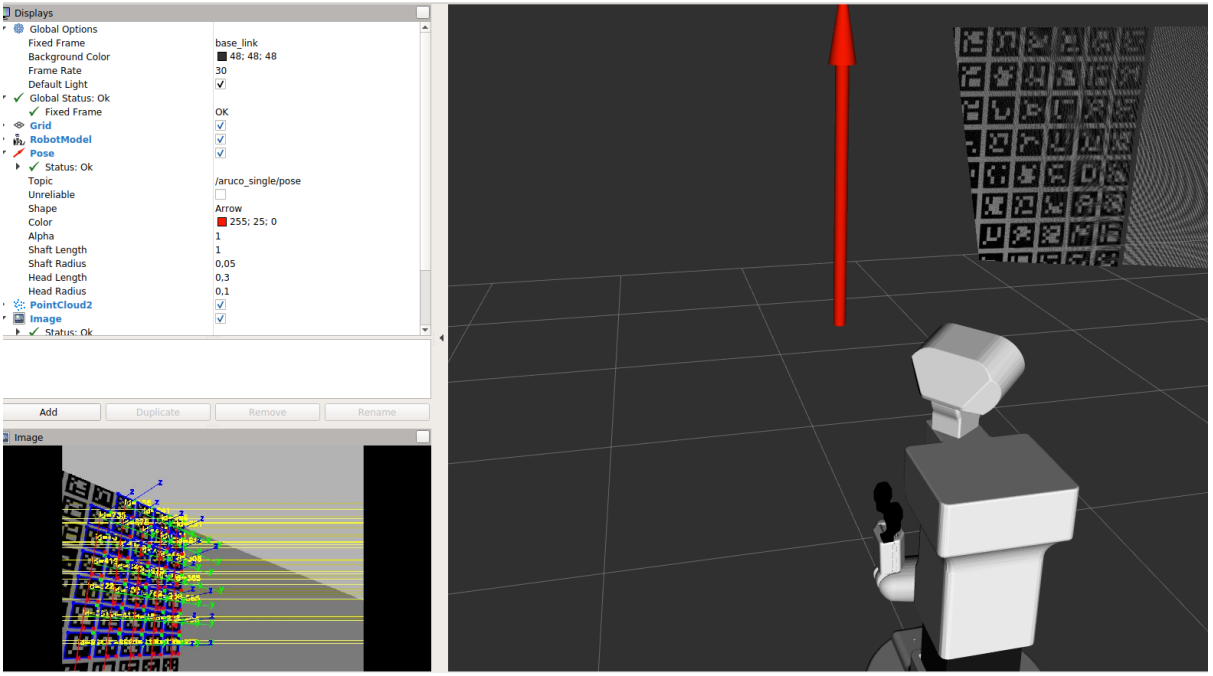
    - Will start the the the different nodes in order to execute the pick up demo
- `rosservice call /pick_gui`
    - Will start the demo
- http://wiki.ros.org/Robots/TIAGo/Tutorials/MoveIt/Pick_place

OpenCV

Corner Detection

- `rosrun tiago_opencv_tutorial corner_detection`
- Uses Harris and Shi-Tomasi operations

ArUco marker detection



- `roslaunch tiago_gazebo tiago_gazebo.launch public_sim:=true robot:=steel world:=tutorial_office gzpose:="-x -3.5 -y -0.85 -z 0 -R 0 -P 0 -Y -3.0"`
    - To spawn in front of the Aruco table
- `roslaunch tiago_aruco_demo detector.launch`
    - To detect the aruco elements
    - `roslaunch tiago_aruco_demo detector.launch markerId:=<marker_ID>`
        - To detect specific aruco element

Person detection

- http://wiki.ros.org/Robots/TIAGo/Tutorials/PersonDetection